SED (CITS2220) Assignment 1 (Design)

**Aim:** The aim of this assignment is to test your ability to design a medium sized software system and to demonstrate your ability to manage and present clearly a portfolio of design documents and supporting information.

**Task:** Perform Requirements Analysis and develop a System Design for a computer system to support the membership management for a community organization with a central headquarters but a dynamic range of local branches. The system will be called the Dynamic Branch Membership System (DBMS). The requirements definition for this system is given in the DBMS Requirements Definition reproduced below. Use a web page interface to give easily navigable access to all your analysis and design documents, including all working versions. As well as the analysis and design documents, you should document the rationale for the design decisions you make and also keep and display a log file (listing date, task, people involved, time taken) for all work undertaken on this project.

**Due Date:** Due 4pm Tuesday 21st April (week 8).
- **Part A** is to be submitted electronically using cssubmit by 4pm on the due date.
- **Part B** is to be handed to the Computer Science Office 1.31A by 4pm on the due date.
Please ensure that you keep a backup copy of both parts for yourself before submission. If your diagrams in Part B are hand-drawn, then you can scan all the pages of your assignment into a pdf file using any photocopier in a UWA library.

All late submissions will be penalized in accordance with the School of CSSE policy on late submission: “an item submitted after the deadline but within 24 hours of the original deadline, will have the mark reduced by the 20% penalty. If submitted between 24 hours and 48 hours late, a penalty of 40% applies, and so on, until the penalty reaches 100% after 4 days. You may choose to submit late assessment after 4 days, but it will contribute nothing to the unit’s overall assessment.” For the full policy see [http://www.csse.uwa.edu.au/departmental/publications/latesubmission.html](http://www.csse.uwa.edu.au/departmental/publications/latesubmission.html)

**Working in Pairs:** This assignment is intended to be undertaken by students working in pairs. You are allowed to work alone on it but improved group work is an aim of this unit and so is encouraged. If you choose to work alone then we will expect full answers to part B but you will not have to complete quite so much of part A.
Otherwise, I will assume that you wish to complete the assignment alone. Both partners are required to contribute equal effort to the assignment — see the Unit Coordinator as soon as possible (well before the deadline) if you believe this is not the case. Please feel free to post to help2220 (the discussion forum) in order to line up a partner.

**Project Planning** This assignment is worth 15% of the total marks for CITS2220. You are expected to spend 20 to 25 hours on this assignment, including background reading and making all versions of the deliverables. A cover sheet, detailing the marking criteria is attached. You must submit this cover sheet with your assignment.
The work you submit for this assignment must be your own (either the designated pair or the individual). In accordance with the School of CSSE Plagiarism policy “Any contribution from others … must be acknowledged as part of the submitted work. Students must inform the unit
Assignment 1 Deliverables

**Part A (20%):** Submit your assignment web page and the linked documents you have developed. Your web page should support easy navigation and viewing of all your documents. In particular, the web page should provide access to

1. a requirements analysis document (RAD),
2. a system design document (SDD),
3. documented rationale for the design decisions you have made,
4. a work log giving the date, task, personnel and time spent on all activities for this assignment, and
5. a list of all reference sources you used for the assignment (including help from other people, templates, books, articles and all web material).

As you may store your document files in different subdirectories, you will need to ensure that the directory structure is retained when your assignment is marked. I suggest that you zip the entire directory into a zip file and submit that zip file. Please try to download your own submission and check that all the files are included in the zip file.

Part A does not have to be complete. It is only worth 20% and you should only spend an appropriate amount of time on it. This applies in particular to students working alone. Part A should, however, give an outline of what the complete documentation would look like, and should contain full details of the parts relevant to part B. The absence of other aspects (such as for example other use cases, other classes, other sequence diagrams) should be noted at the appropriate place.

**Part B (80%):** Submit a completed project cover sheet (attached) together with paper copies of one to two page answers to each of the following 8 questions, together with a paper copy of your list of reference sources (see item 5 of Part A above). Your answers should be based on the full Requirements Analysis and System Design documents you develop in Part A. Written submissions should be no more than two A4 pages per question including diagrams. Include references to the documents of Part A where relevant.

1. Present an overview of the DBMS system using a UML use case diagram together with some brief explanatory text.
2. Draw a conceptual level UML class diagram for the main objects identified by your analysis. Include multiplicities and roles of object associations in your diagram but do not show attributes and operations of the objects yet.
3. For each of the objects identified in question 2, list the attributes of that object, and its operations. Include brief explanatory comments on the meaning of your attribute and operation names.
4. The “New Member” use case (3.1 in the Requirements Definition) involves interactions between several actors and objects (including membership database and email servers). Draw a UML sequence diagram together with some brief explanatory text to define the interactions between objects which occur in this use case.
5. List up to 5 design constraints for this system in order of priority. Explain your reason for each constraint (e.g. reference to the Req.Defn. document or to related systems) and the priority you have given it.
6. Outline a subsystem decomposition for the DBMS system, identifying the services provided

coordinator if their work is done jointly or borrows heavily from others. Failure to do so is plagiarism.” This includes the use of templates, text book examples and all information from the web. For the full policy see http://www.csse.uwa.edu.au/departmental/publications/policy.on.plagiarism.html
by each of your subsystems to others (see B&D p228-246). Use a diagram as well as explanatory text.

7. Make a list of Boundary (or System Admin) Use Cases you will need for the DBMS system (see B&D p277-278). Choose one of these use cases, and give a textual description of it. Use the template provided in the handouts for lecture 2.

8. Identify up to 3 exceptions for the DBMS system (see B&D p278-279) and outline how each exception will be handled by the system.

Resources
Sample templates for requirements analysis and system design documents can be found in RADtmpl.html and SDDtmpl.html on http://undergraduate.csse.uwa.edu.au/units/CITS2220/se.resourcespage.htm

Sample web pages for managing SW design documents can be found online at http://undergraduate.csse.uwa.edu.au/units/CITS2220/practicals/James1.htm for the James system. Please note, however, that your assignment will be a much smaller development effort than that of the B&D examples and not all sections of the document templates or web page interfaces will be relevant for your example.

Chapters 5, 6 and 7 of Bruegge and Dutoit’s book Object Oriented Software Engineering gives more details about each of the deliverables for Part B.

Marking criteria for the assignment are detailed on the project cover sheet. Make sure you read these carefully to determine what is required for each part of the submission.

Important Advice:
1) do not add more functionality to your system than is necessary without a good reason: there are more important things to spend your time on;
2) note the 20%-80% division of marks between tasks and expend your effort accordingly.

Du Huynh, CITS2220 Unit Co-ordinator, March 2009
DBMS: Dynamic Branch Membership System
(Requirement Definition Document)

1 System Objectives
The Dynamic Branch Membership System (DBMS) is a software application to assist in the task of managing the membership of a community organization which has a national HQ but a variable number of local branches.

2 System Context
1. DBMS will provide a central server program to assist with managing branch membership for a community organization “Wetland Watch” (WW). DBMS helps with the membership fee records, allocation of new members to branches, and the formation of new branches.
2. The system communicates with four separate existing systems: a commercially run credit card payment service; a commercially run online Australian address, postcode and distance service; (via a local network) the current national membership records database server for “Wetlands Watch”; an email server.
3. The existing membership database contains contact information, membership payment records and branch details. However, it is not accessible from the Internet.
4. Administration staff are able to set up DBMS, connect it to the other server programs and input current branch details.
5. Each branch has a secretary (who is a member of WW) and each member of WW will be (eventually) allocated to a branch. There is also national secretary of WW who needs to be informed (via email) of certain membership decisions.
6. Branch secretaries, ordinary members, and prospective members will only access DBMS through a secure web facility.

3 Functional Requirements
3.1) New Member
1) A public web site allows interested members of the public to choose to become members of the WW.
2) The user selects the “become a member” option and goes through a process of entering personal information followed by a credit card payment process. There is only one standard amount to be charged which is annual membership for an individual.
3) Credit card membership payments are handled by a separate online service provided by a bank. DBMS informs the service of the amount to charge and then allows the user to interact with the service directly (and securely). If the amount is charged successfully then this service informs DBMS and allows DBMS to continue interacting with the user.
4) Membership details are recorded in the membership database.
5) The new member is welcomed to WW and then informed that they are expected to sign up to belong to local branch. DBMS uses a separate commercially available geographic online service
to display to the new member a list of the 10 local branches of WW which are closest in
distance to the user’s home address.

6) The user is asked to select one branch to belong to. They are informed (on the web page) that
the branch secretary will contact them in the next few days. The member is also sent a
confirmation email which also gives them a temporary password to use to allow log in to the
DBMS as a member.

7) An email is automatically sent to the selected branch secretary informing them of the existence
of the new member. Contact details of the new member are sent to the secretary.

8) Reminder emails are sent every week to the branch secretary about the need to contact these
“pending branch members” until the branch secretary logs in to DMBS to confirm that they
have contacted the new member and accepted them.

9) The new member is then sent an email confirming that they now belong to that branch.
Alternatively, if the branch secretary reports that the branch is not happy to accept the new
member then an email to that effect is sent. An email is also sent to the national secretary of
WW.

3.2) Edit existing member

1) At any time an existing member of WW can log in to the DBMS (via a web page) to view or
edit their contact details. Alternatively an admin staff member can log in to edit the details of
any member.

2) They can also search, browse and view alternative branches to which they could belong.

3) If desired they can request to be transferred to another branch. In that case their current branch
allocation is canceled. The member is added to the list of pending branch members for the other
branch and (as above) will wait for contact and confirmation from the branch secretary.

3.3) Create new branch

1) A member can log in to DBMS to start a new branch. Alternatively, administrative staff can do
this for them.

2) The member specifies the name and location of the branch. The member him/herself will be the
new branch secretary.

3) An email is sent to the national secretary requesting approval of the formation of the new
branch.

4) Approval (of the set up of the new branch) is indicated to DBMS by administrative staff.

5) If approved, the new branch secretary and members in nearby branches are informed of the
creation of the new branch by email (automatically generated by DBMS).
3.4) Edit branch

1) The branch secretary can log in at any time to edit details of the branch.

2) S/he can change branch location or change the branch secretary to be another member of the branch. Branch members and the national secretary are informed by email.

3) The branch secretary can view and then accept or deny any pending branch members. They will be informed by DMBS via email.

4) The branch can be closed down. Members are informed by email and asked to log in at their convenience to choose a new branch.

3.5) Renewal Reminders

5) DBMS will automatically send emails to members to remind them a few weeks before membership renewal is due.

4 Quality Requirements

4.1 Performance
4.1.1 DBMS shall show no visible deterioration in response time as the number of users of the system increases.
4.1.2 DBMS shall require a reasonably small amount of memory so that enough of it is permanently resident on the server to provide quick service.
4.1.3 DBMS shall load as quickly as comparable productivity tools on whatever environment it is running in.
4.1.4 DBMS will respond to client web activity in a timely and convenient way.

4.2 Reliability
4.2.1 DBMS shall be available for use as much as comparable productivity tools.

4.3 Usability
4.3.1 DBMS shall provide a standard style of user interface so that users do not have to learn a new style of interaction.
4.3.2 Users will be able to understand the layout and options of the DBMS user interface.
4.3.3 Notification and email messages generated by DBMS shall be clear, succinct, and polite and free of jargon.

4.4 Portability
4.4.1 DBMS will be implemented on a platform that allows easy re-hosting on different hardware and OS.

4.5 Modifiability
4.5.1 DBMS will be implemented using modern programming practices that maximize the maintainability and reusability of designs and code.
4.5.2 DBMS will be implemented in such a way that alternative authentication, geographical,
payment, database and email server programs could be used easily without affecting the logic of the design.

5 Future Requirements

5.1 Facility for branches to use DBMS to coordinate their group meetings.
5.2 Support for SMS and other means of contact besides email.
CITS2220 Assignment 1 Cover Sheet 2009

<table>
<thead>
<tr>
<th>Student 1</th>
<th>Student 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project group number</strong> (if applicable)</td>
<td></td>
</tr>
<tr>
<td><strong>Family Name</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Given Names</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Student Number</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Signature</strong></td>
<td></td>
</tr>
</tbody>
</table>

**TASK A**

**ONLINE SUBMISSION OF ALL DESIGN DOCUMENTS**

<table>
<thead>
<tr>
<th><strong>Ease of use</strong></th>
<th>20/100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document collection and its web page interface has clear structure and is easy to navigate</td>
<td>/5</td>
</tr>
<tr>
<td><strong>Completeness</strong>: 1. RAD, 2. SDD, 3. rationale document, 4. work log, and 5. references. All parts submitted and accessible from the assignment web page.</td>
<td></td>
</tr>
<tr>
<td><strong>Readability</strong>: Documents written in clear prose; Succinct style with irrelevant sections of the document templates removed; Appropriate use of diagrams; documents are easy to navigate; duplication of information is minimized.</td>
<td>/5</td>
</tr>
</tbody>
</table>

**TASK B**

**PAPER SUBMISSION OF DESIGN SUMMARY**

| **Q1. Use case diagram**: appropriate choice of actors and use cases which covers all the functionality of the given requirements definition but does not invent any new functionality; uses the client's language and not new or ambiguous terms; boundary between this system and external systems clearly identified and justified | 80/100 |
| **Q2. Conceptual Class Model**: Significant relationships between objects identified (with reference to Req.Defn); Reasonable roles and multiplicities identified (with reference to Req.Defn). | /5 |
| **Q3. Object Model**: Attributes and operations identified for each object (with reference to Req.Defn); Uses the client's language and not new or ambiguous terms; Names of attributes and operations explained; | /10 |
| **Q4. Sequence Diagram**: participating actors identified; main interactions from the use case included; correct use of UML sequence chart syntax; | /10 |
| **Q5. Design Constraints**: Constraints are linked to Req.Defn; Justification given for inclusion of each constraint and its priority; | /10 |
| **Q6. Subsystem Decomposition**: services provided by each subsystem listed; justification of good design such as low coupling, high cohesion; show that decomposition satisfies system design criteria (see Q5); | /15 |
| **Q7. Boundary Use Cases**: list of use cases identified with justification from Req.Defn; use case template correctly completed for a significant case; normal and exceptional behaviour specified; | /10 |
| **Q8. Exceptions**: 3 exceptions identified are significant for the DBMS; sensible mechanism for handling each exception is given | /10 |

**MAJOR STRENGTHS**

**MAJOR WEAKNESSES**

**MARKER:**

**TOTAL MARK:**